UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/602,126	06/24/2003	Cliff M. R. Don	13768.444	1587

47973 7590 04/06/2007

WORKMAN NYDEGGER/MICROSOFT
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UT 84111

EXAMINER	
MORRISON, JAY A	
ART UNIT	PAPER NUMBER
2168	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	04/06/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	**Application No.** 10/602,126	**Applicant(s)** DON ET AL.
	Examiner Jay A. Morrison	**Art Unit** 2168

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 24 January 2007.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-5,7-18,21-28 and 30-38 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-5,7-18,21-28 and 30-38 is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on 24 June 2003 is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

Remarks

1. Claims 1-5,7-18,21-28,30-38 are pending.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

3. Claims 1-5,7-18,21-28,30-38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gombocz et al. ('Gombocz' hereinafter) (Publication Number 2002/0156792) in view of Wong et al. ('Wong' hereinafter) (Patent Number 6,889,229).

As per claim 1, Gombocz teaches

In a multi-tier server system that includes a back end server at a first tier and a plurality of additional servers at a middle tier, each additional server using multiple types of data objects that must be defined on the plurality of additional servers before the data objects can be used by the plurality of middle tier servers, a method for deploying one or more data types from the back end server to the plurality of middle tier servers in a manner that maintains consistency and compatibility in the definitions of the data types and in code associated with each data type as stored on each middle tier server in the system, the method comprising: (see abstract and background)

an act of creating a special table in a database of the back end server, the special table including one or more fields for storing data identifying data types used by the plurality of middle tier servers and corresponding code for enabling use of each of the data types, the database of the back end server acting as a repository for each data type used by any of the plurality of middle tier servers, ... and the corresponding code required to enable use of the data types by the plurality of middle tier servers; (data type, access and structure definition tables, application type, database definition generator, application/database definition generator enables detection of structural and functional information necessary for standardization, paragraph [0194]-[0195])

an act of identifying a data type to be deployed from the back end server to the-one or more of the plurality of middle tier servers; (application/database definition generator enables detection of structural and functional information necessary for standardization, paragraph [0195])

an act of obtaining an extended assembly that corresponds to the data type to be deployed, the extended assembly including data obtained from the special table, including data identifying the data type, one or more definitions of the data type, and the code for enabling processing of data corresponding to the data type; (application/database definition generator enables detection of structural and functional information necessary for standardization, paragraph [0195])

and an act of transmitting the extended assembly to the one or more middle tier servers of the plurality of middle tier servers in the multi-tier system such that the data type, as transmitted to and received by the one or more of the plurality of middle tier servers in the multi tier system, is consistent and compatible with a data type of the same kind stored on other middle tier servers in the system. (information interchange with components, paragraph [0196])

Gombocz does not explicitly indicate "and the back end server acting as a single and centralized source from which each of the plurality of middle tier servers obtains all data types used by any other of the plurality of middle tier servers".

However, Wong discloses "and the back end server acting as a single and centralized source from which each of the plurality of middle tier servers obtains all data types used by any other of the plurality of middle tier servers" (master site, column 6, lines 14-20).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Gombocz and Wong because using the steps of "and the back end server acting as a single and centralized source from which each of the plurality of middle tier servers obtains all data types used by any other of the plurality of middle tier servers" would have given those skilled in the art the tools to improve the invention by maintain database types across networks. This gives the user the advantage of not having inconsistent types.

As per claim 2, Gombocz teaches

an act of creating logic modules in the-one or more of the plurality of middle tier servers that enable the one or more of the plurality of middle tier servers to query for the extended assembly. (paragraph [0065])

As per claim 3, Gombocz teaches

the back end server includes a relational database. (paragraph [0032])

As per claim 4, Gombocz teaches

the back end server comprises an SQL server. (paragraph [0032])

As per claim 5, Gombocz teaches

the one or more of the plurality of middle tier servers includes an email server. (paragraph [0137])

As per claim 7, Gombocz teaches

the act of identifying the data type to be deployed includes determining that the one or more of the plurality of middle tier servers has requested the extended assembly, since the one or more of the plurality of middle tier servers are not yet enabled for the data type. (paragraph [0065])

As per claim 8, Gombocz teaches

an act of adding a new middle tier server to the multi-tier system, and wherein the new middle tier server comprises the one or more of the plurality of middle tier servers that has requested the extended assembly. (paragraph [0134])

As per claim 9, Gombocz teaches

an act of creating one or more object tables that are linked to the special table and that include additional information defining the data type to be deployed, such that the extended assembly also includes the additional information. (paragraph [0207])

As per claim 10, Gombocz teaches

In a multi-tier server system that includes a back end server at a first tier and a plurality of additional servers at a middle tier, each additional server using multiple types of data objects that must be defined on the plurality of additional servers before the data objects can be used by the plurality of middle tier servers, a method for deploying one or more data types from the back end server to the plurality of middle tier servers in a manner that maintains consistency and compatibility in the definitions of the data types and in code associated with each data type as stored on each middle tier server in the system, the method comprising: (see abstract and background)

an act of modifying a special table in a database of the back end server, the special table including one or more fields for storing data that identifies data types used by the plurality of middle tier servers and includes corresponding code for enabling use of each of the data types, the database of the backend server being a repository for each data type used by any of the plurality of middle tier servers, ... and the corresponding code required to enable use of the data types by the plurality of middle tier servers, the act of modifying including at least one of modifying the stored data within the one or more fields and adding new stored data to the one or more fields; (data type, access and structure definition tables, application type, database definition generator, application/database definition generator enables detection of structural and functional information necessary for standardization, paragraph [0194]-[0195])

an act of identifying a data type to be deployed from the back end server to one or more of the plurality of middle tier servers; (application/database definition generator enables detection of structural and functional information necessary for standardization, paragraph [0194]-[0195])

an act of obtaining an extended assembly that corresponds to the data type to be deployed, the extended assembly including at least one of the modified stored data and the new stored data as obtained from the special table, including data identifying the data type, and the executable code that, when executed, enables the one or more middle tier servers to process the modified stored data or the new stored data associated with the data type; (application/database definition generator enables detection of structural and functional information necessary for standardization, paragraph [0194]-[0195])

and an act of transmitting the extended assembly to the one or more middle tier servers of the plurality of middle tier servers in the multi-tier system such that the data type as transmitted to and received by the one or more of the plurality of middle tier servers in the multi tier system is consistent and compatible with a data type of the same kind stored on other middle tier servers in the system. (information interchange with components, paragraph [0196])

Gombocz does not explicitly indicate "and the back end server acting as a single and centralized source from which all middle tier servers obtain all data types used by any other of the plurality of middle tier servers".

However, Wong discloses "and the back end server acting as a single and centralized source from which all middle tier servers obtain all data types used by any other of the plurality of middle tier servers" (master site, column 6, lines 14-20).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Gombocz and Wong because using the steps of "and the back end server acting as a single and centralized source from which all middle tier servers obtain all data types used by any other of the plurality of middle tier servers" would have given those skilled in the art the tools to improve the invention by maintain database types across networks. This gives the user the advantage of not having inconsistent types.

As per claim 11, Gombocz teaches

an act of determining which of the plurality of middle tier servers should be sent the extended assembly. (paragraph [0065])

As per claim 12, Gombocz teaches

determining which of the plurality of middle tier servers should be sent the extended assembly comprises the acts of: sending data associated with the data type to the one or more of the plurality of middle tier servers; (paragraph [0065])

and receiving one or more requests for the extended assembly from the one or more of the plurality of middle tier servers upon the one or more of the plurality of middle tier servers identifying that the data associated with the data type cannot be processed at the one or more of the plurality of middle tier servers. (paragraph [0207])

As per claim 13, Gombocz teaches

the back end server includes a relational database. (paragraph [0032])

As per claim 14, Gombocz teaches

the back end server comprises an SQL server. (paragraph [0032])

As per claim 15, Gombocz teaches

the one or more of the plurality of middle tier servers includes an email server. (paragraph [0137])

As per claim 16, Gombocz teaches

the act of modifying includes adding new stored data corresponding to a new data type not previously enabled in the multi-tier system prior to adding the new stored data. (paragraph [0070]

As per claim 17, Gombocz teaches

In a multi-tier server system that includes a back end server at a first tier and a plurality of additional servers at a middle tier, each additional server using multiple types of data objects that must be defined on the plurality of additional servers before the data objects can be used by the plurality of middle tier servers, a method for deploying one or more data types from the back end server to the plurality of middle tier servers in a manner that maintains consistency and compatibility in the definitions of the data types and in code associated with each data type as stored on each middle tier server in the system, the method comprising: (see abstract and background) an act of adding a new middle tier server to the multi-tier system, the new middle tier server being configured to utilize extended assemblies that are obtained from the back end server, the back end server acting as a repository for storing each data type used by any of the plurality of middle tier servers, ... and corresponding code required to enable use of the data types by the plurality of middle tier servers, the extended assemblies being configured to enable the use of one or more data types that are defined by data and enabled by executable code that is contained in the extended assemblies; (data type, access and structure definition tables, application type, database definition generator, application/database definition generator enables detection of structural and functional information necessary for standardization, paragraph [0194]-[0195])

an act of determining which of the one or more data types are to be deployed from the back end server to the new middle tier server, wherein the act of determining is based at least in part on a request by the new middle tier server for data to enable use of one or more data types; (application/database definition generator enables detection of structural and functional information necessary for standardization, paragraph [0194]-[0195])

an act of obtaining one or more extended assemblies corresponding to the one or more data types that have been determined to be deployed, each of the one or more extended assemblies including data and executable code obtained from a special table stored in a database of the back end server, the special table including one or more fields for storing data identifying data types and corresponding code for processing data associated with the data types; (application/database definition generator enables detection of structural and functional information necessary for standardization, paragraph [0194]-[0195])

and an act of transmitting, to the new middle tier server, the one or more extended assemblies that correspond to the one or more data types that have been determined to be deployed, such that the one or more data types as transmitted to, and received by, the new middle tier server are consistent and compatible with one or more data types of the same kind on other middle tier servers in the system, and which were received by the other middle tier servers from the back end server. (information interchange with components, paragraph [0196])

Gombocz does not explicitly indicate "and the back end server acting as a single and centralized source from which all middle tier servers obtain all data types used by any other of the plurality of middle tier servers".

However, Wong discloses "and the back end server acting as a single and centralized source from which all middle tier servers obtain all data types used by any other of the plurality of middle tier servers" (master site, column 6, lines 14-20).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Gombocz and Wong because using the steps of "and the back end server acting as a single and centralized source from which all middle tier servers obtain all data types used by any other of the plurality of middle tier servers" would have given those skilled in the art the tools to improve the invention by maintain database types across networks. This gives the user the advantage of not having inconsistent types.

As per claim 18, Gombocz teaches

the act of determining is further based at least in part on the new middle tier server identifying what other data types are supported, and identifying that the one or more data types to be deployed are not supported at the new middle tier server. (paragraph [0134])

As per claim 21, Gombocz teaches

In a multi-tier server system that includes a back end server at a first tier and a plurality of additional servers at a middle tier, each additional server using multiple types of data objects that must be defined on the plurality of additional servers before the data objects can be used by the plurality of middle tier servers, a method for deploying one or more data types from the back end server to the plurality of middle tier servers in a manner that maintains consistency and compatibility in the definitions of the data types and in code associated with each data type as stored on each middle tier server in the system, the method comprising: (see abstract and background)

an act of creating a special table in a database of the back end server, the special table including one or more fields for storing data identifying a data type used by the plurality of middle tier servers and corresponding executable code for enabling processing of data associated with the data type, the database of the back end server acting as a repository for storing each data type used by any of the plurality of middle tier servers, ... and the corresponding code required to enable use of the data types by the plurality of middle tier servers; (data type, access and structure definition tables, application type, database definition generator, application/database definition generator enables detection of structural and functional information necessary for standardization, paragraph [0194]-[0195])

a step for deploying the data type from the back end server to one or more of the plurality of middle tier servers, upon request, such that the data type as transmitted to and received by the one or more of the plurality of middle tier servers in the multi-tier server system is consistent and compatible with a data type of the same kind stored on other middle tier servers in the system. (information interchange with components, paragraph [0196]).

Gombocz does not explicitly indicate "and the back end server acting as a single and centralized source from which all middle tier servers obtain all data types used by any other of the plurality of middle tier servers".

However, Wong discloses "and the back end server acting as a single and centralized source from which all middle tier servers obtain all data types used by any other of the plurality of middle tier servers" (master site, column 6, lines 14-20).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Gombocz and Wong because using the steps of "and the back end server acting as a single and centralized source from which all middle tier servers obtain all data types used by any other of the plurality of middle tier servers" would have given those skilled in the art the tools to improve the invention by maintain database types across networks. This gives the user the advantage of not having inconsistent types.

As per claim 22, Gombocz teaches

the step for deploying the data type to the one or more middle tier servers upon request comprises corresponding acts that include: an act of identifying the data type to be deployed based on receipt of the data type at the one or more of the plurality of middle-tier servers, and the one or more of the plurality of middle-tier servers requesting an extended assembly for the data type since the data type cannot be processed at the one or more of the plurality of middle tier servers; (application/database definition generator enables detection of structural and functional information necessary for standardization, paragraph [0194]-[0195])

an act of obtaining an extended assembly that corresponds to the data type to be deployed, the extended assembly including the data from the special table identifying the data type and the executable code for enabling processing of the data associated with the data type; (application/database definition generator enables detection of structural and functional information necessary for standardization, paragraph [0194]-[0195])

and an act of transmitting the extended assembly to the one or more of the plurality of middle tier servers in the multi-tier system that requested the extended assembly. (information interchange with components, paragraph [0196])

As per claim 23, Gombocz teaches

an act of creating logic in the one or more of the plurality of middle tier servers that enables utilization of the extended assembly. (paragraph [0134])

As per claim 24, Gombocz teaches

an act of creating at least one object table in the database of the back end server that includes at least some information defining the data type, and wherein the extended assembly includes the at least some information. (paragraph [0205])

As per claim 25, Gombocz teaches

A computer program product for use in a multi-tier server system that includes a back end server at a first tier and a plurality of additional servers at a middle tier, each additional server using multiple types of data objects that must be defined on the plurality of additional servers before the data objects can be used by the plurality of middle tier servers, the computer program product including one or more computer readable storage media having stored thereon computer-executable instructions for implementing a method for deploying one or more data types from the back end server to the plurality of middle tier servers in a manner that maintains consistency and compatibility in the definitions of the data types and in code associated with each data type as stored on each middle tier server in the system, the method comprising: (see abstract and background) an act of creating a special table in a database of the back end server, the special table including one or more fields for storing data identifying data types used by the plurality of middle tier servers and corresponding code for enabling use of each of the data types, the database of the back end server acting as a repository storing each data type used by any of the plurality of middle tier servers, ... and the corresponding code required to enable use of the data types by the plurality of middle tier servers; (data type, access and structure definition tables, application type, database definition generator, application/database definition generator enables detection of structural and functional information necessary for standardization, paragraph [0194]-[0195])

an act of identifying a data type to be deployed from the back end server to one or more of the plurality of middle tier servers; (application/database definition generator enables detection of structural and functional information necessary for standardization, paragraph [0194]-[0195])

an act of obtaining an extended assembly that corresponds to the data type to be deployed, the extended assembly including data obtained from the special table, including data identifying the data type, one or more definitions of the data type, and the code for enabling processing of data associated with the data type; (application/database definition generator enables detection of structural and functional information necessary for standardization, paragraph [0194]-[0195])

and an act of transmitting the extended assembly to one or more of the plurality of middle tier servers in the multi-tier system such that the data type as transmitted to and received by the one or more of the plurality of middle tier servers in the multi tier system is consistent and compatible with a data type of the same kind stored on other middle tier servers in the system. (information interchange with components, paragraph [0196])

Gombocz does not explicitly indicate "and the back end server acting as a single and centralized source from which all middle tier servers obtain all data types used by any other of the plurality of middle tier servers".

However, Wong discloses "and the back end server acting as a single and centralized source from which all middle tier servers obtain all data types used by any other of the plurality of middle tier servers" (master site, column 6, lines 14-20).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Gombocz and Wong because using the steps of "and the back end server acting as a single and centralized source from which all middle tier servers obtain all data types used by any other of the plurality of middle tier servers" would have given those skilled in the art the tools to improve the invention by maintain database types across networks. This gives the user the advantage of not having inconsistent types.

As per claim 26, Gombocz teaches

the method further includes an act of creating logic modules in the one or more of the plurality of middle tier servers that enable the one or more of the plurality of middle tier servers to query for the extended assembly. (paragraph [0207])

As per claim 27, Gombocz teaches

the back end server includes an SQL server. (paragraph [0032])

As per claim 28, Gombocz teaches

the one or more of the plurality of middle tier servers includes an email server. (paragraph [0137])

As per claim 30, Gombocz teaches

the act of identifying the data type to be deployed includes determining that the one or more of the plurality of middle tier servers has requested the extended assembly, since the one or more of the plurality of middle tier servers are not yet enabled for the data type. (paragraph [0134])

As per claim 31, Gombocz teaches

the method further includes an act of adding a new middle tier server to the multi-tier system, and wherein the new middle tier server comprises the one or more of the plurality of middle tier servers that has requested the extended assembly. (paragraph [0207])

As per claim 32, Gombocz teaches

an act of creating one or more object tables that are linked to the special table and that include additional information defining the data type to be deployed, and wherein the extended assembly also includes the additional information. (paragraph [0205])

As per claim 33, Gombocz teaches

the method further includes modifying at least one of the special table and the one or more object tables. (paragraph [0207])

As per claim 34, Gombocz teaches

the extended assembly is a single data structure that includes all the data required to enable the one or more of the plurality of middle tier servers to use the data type. (paragraph [0134])

As per claim 35, Gombocz teaches

the one or more of the plurality of middle tier servers have limited program code means to process data associated with less than all of the data types in the multi-tier system, and the back end server has all program code means to process any data associated with all of the data types in the multi-tier system. (paragraph [0207])

As per claim 36, Gombocz teaches

the one or more of the plurality of middle tier servers are only equipped to recognize and process data objects associated with a particular data type when program code means comprising executable machine code of the extended assembly for the particular data type has been received from the back end server and installed at the one or more of the plurality of middle tier servers. (paragraph [0205])

As per claim 37, Gombocz teaches

At a middle tier server in a multi-tier database server system that includes a back end database server at a first tier and a plurality of additional database servers at a middle tier, wherein the plurality of middle tier servers are configured to process data corresponding to data types defined by the back end server at the first tier, a method for deploying one or more data types from the back end server to the plurality of middle tier servers in a manner that maintains consistency and compatibility in the definitions of the data types and in code associated with each data type in the multi-tier database server system, the method comprising: (see abstract and background)

an act of receiving at a middle tier server one or more data objects from a back end server, the one or more received data objects associated with at least one data type; (information interchange with components, paragraph [0196])

an act of initiating one or more processing functions for the one or more received data objects associated with the at least one data type; (application/database definition generator enables detection of structural and functional information necessary for standardization, paragraph [0194]-[0195])

an act of identifying that the at least one data type of the one or more data objects is not recognized, such that the initiated one or more initiated processing functions have failed at the middle tier server; (application/database definition generator enables detection of structural and functional information necessary for standardization, paragraph [0194]-[0195])

an act of pulling one or more extended assemblies corresponding to the at least one data type from the back end server, wherein the one or more extended assemblies were generated by the back end server in response to a request from the middle tier server to pull the one or more extended assemblies, wherein the back end server acts as a repository for each data type used by any of the plurality of middle tier servers; (information interchange with components, paragraph [0196])

and an act of processing the one or more data objects associated with the at least one data type using the pulled one or more extended assemblies, wherein the middle tier server successfully recognizes the at least one data type, and successfully processes the one or more received data objects associated with the at least one data type. (application/database definition generator enables detection of structural and functional information necessary for standardization, paragraph [0194]-[0195])

Gombocz does not explicitly indicate "and as a single and centralized source from which each of the plurality of middle tier servers obtains all extendable assemblies corresponding to all data types used by any other of the plurality of middle tier servers".

However, Wong discloses "and as a single and centralized source from which each of the plurality of middle tier servers obtains all extendable assemblies corresponding to all data types used by any other of the plurality of middle tier servers" (master site, column 6, lines 14-20).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Gombocz and Wong because using the steps of "and as a single and centralized source from which each of the plurality of middle tier servers obtains all extendable assemblies corresponding to all data types used by any other of the plurality of middle tier servers" would have given those skilled in the art the tools to improve the invention by maintain database types across networks. This gives the user the advantage of not having inconsistent types.

As per claim 38, Gombocz teaches

the one or more pulled extended assemblies comprise computer executable instructions that, when executed at the middle tier server, cause one or more processors at the middle tier server to format the one or more data objects so that the one or more data objects can be processed. (paragraph [0205])

Response to Arguments

4. Applicant's arguments with respect to claims 1-5,7-18,21-28,30-38 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

5. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP

§ 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

The prior art made of record, listed on form PTO-892, and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jay A. Morrison whose telephone number is (571) 272-7112. The examiner can normally be reached on M-F 8-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim Vo can be reached on (571) 272-3642. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

TIM VO
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Jay Morrison
TC2100

Tim Vo
TC2100